Bug or Feature?

Cameron Browne, Queensland University of Technology (QUT)

This article explores the relevance of the 'bug or feature' concept found in computer programming to the process of game design. Several examples are presented of successful games with either apparent bugs that proved beneficial on closer analysis, or actual bugs whose solution provided worthwhile emergent benefits. Game design is posed as a bug fixing process.

1 Introduction

THERE is an old joke among computer programmers that when a customer complains about a bug[1] in a piece of software, it is sometimes easiest to just describe the resulting behaviour as a 'feature' of the program. For example, an error that inadvertently deletes a non-critical database every millionth entry might be described as a 'memory saving feature'.

This analogy can be extended to the practice of game design, which often starts with an initial idea for a desired mechanism or behaviour and some preferred set of equipment, followed by an iterative process of identifying bugs in the design and fixing them, hopefully improving the game with each iteration. In this context, a *bug* refers to some undesirable behaviour and a *feature* refers to some desirable behaviour, resulting from the interaction between the rules and the equipment.

This article explores two aspects of the bug/feature dichotomy relative to game design; bugs that are actually features and bugs that can be turned into features. Sometimes it is not clear whether a particular design aspect is a bug or a feature, until the situation is studied in greater depth.

We are not interested in mere problem solving here; the design of *any* game could be described as one long process of bug fixing. Instead, we are interested in bugs that produce some emergent and unexpected benefits, either through side effects or through their solution, that add some significant feature to the game. The best examples play off the detrimental behaviour of the bug to produce some beneficial result.

2 Bug or Feature

This first set of examples includes games that contain apparent design bugs that have turned out to be positive features (or can be viewed as such), without modification.

2.1 Mambo

Mambo[2] is a tile placement game for two players, Red and Blue, who take turns placing one of the Mambo tiles shown in Figure 1 (left) in any orientation to match at least one adjoining tile. The aim is to kill an enemy group by stopping it from further growth. For example, Red has killed the central Blue group to win in Figure 1 (right).

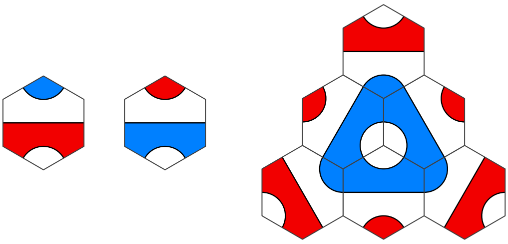

Figure 1. The Mambo tiles and a Red win.

The rules for Mambo were initially simpler and required only that players close an enemy group. However, this initial rule set had a problem in that players could create unplayable *null points* that no tile placement could match, such as the point marked × in Figure 2, and thus protect their groups from closure to avoid defeat *ad infinitum*.

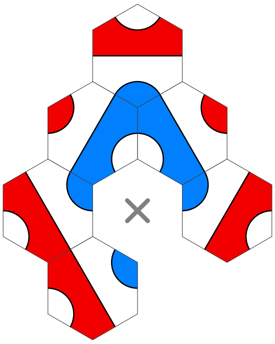

Figure 2. No tile can be played at this null point.

[1]The term *bug* refers here to an error in a program or mechanical system that produces unexpected behaviour.
[2]http://www.cameronius.com/games/mambo/

Browne, C., 'Bug or Feature?', *Game & Puzzle Design*, vol. 2, no. 1, 2016, pp. 46–53. © 2016

This posed a serious problem, as such null points cannot be eliminated from the game without the addition of cumbersome rules for handling special cases. However, a change of perspective saved the day; simply changing the winning condition from ‘close an enemy group’ to ‘stop an enemy group from further growth’ solved this problem elegantly. Embracing null points as a part of the game added strategic depth and made wins much more likely to occur, resulting in a nicely balanced yet aggressive game in which both players are typically only one move away from defeat. Thus, an apparent bug proved to be a key feature of the game, with a simple change of perspective (and slight rule tweak).

2.2 Blue

Blue is a tile placement game for three players, denoted White, Blue and Grey, who take turns placing one of the tiles shown on the left of Figure 3 on a square grid each turn, such that edge colours match adjacent neighbours. Players score 10 points for each completed line of their colour and 1 point for each completed circle of their colour.[3] Figure 3 (right) shows a game in progress, in which White has 11 points, Blue has 23 points and Grey has 31 points.

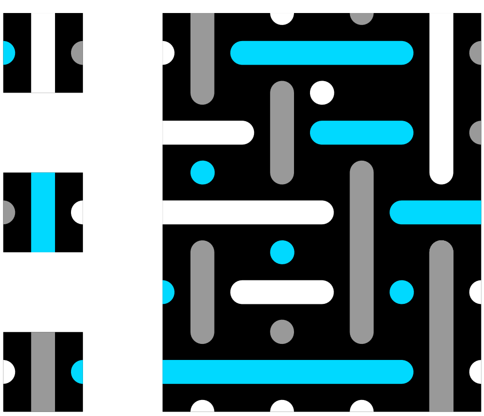

Figure 3. The Blue tiles and an example position.

The tiles were initially intended to achieve fully packed tessellations, such as that shown in Figure 3. However, a bug emerged on the very first playtest, when spaces with two adjacent sides of the same colour proved unplayable, similar to null points in Mambo. For example, no tile can legally be played at the top right space in Figure 4, marked $\times$. This problem of unplayable spaces would have plagued most designers of tile placement games.

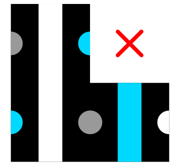

Figure 4. An unplayable space.

However, it soon became apparent that such unplayable spaces add an important tactical element to the game, as they allow players to judiciously block their opponents from completing point-scoring lines or dots.[4] This feature proved so important to the game that additional tokens are provided in the published set, for players to explicitly mark such unplayable spaces as they occur. This is another case of an apparent bug that proved to be a key feature of the game.

2.3 Margo

Margo is a 3-dimensional version of Go, in which marbles played on a square grid stack upwards [4]. The capture rules are similar to Go; groups with no *freedom* (adjacent empty board holes) are captured and removed. For example, Figure 5 (left) shows a black group in *atari*[5] with one freedom remaining at the cell marked $+$.

Move 1 (right) removes this freedom and captures the group, but the question now arises as to what should happen to the pinned black piece marked **z**. Should this piece also be removed? If so, which of the pinning white pieces should drop down to fill the gap that its removal would leave? And how then should captured pieces that are pinned from all angles and hidden from the player’s view be removed?

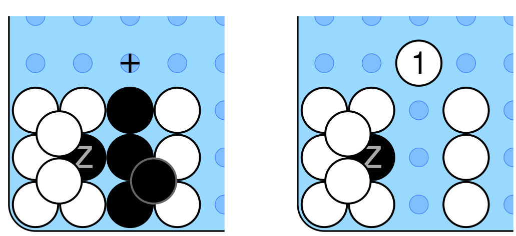

Figure 5. A capture in Margo.

The solution was easy: just leave such pinned pieces where they are, to remain active in the game as *zombies*. This soon proved to be one of the most interesting aspects of the game, with many tactical and strategic implications. For example, zombies allow groups to live safely with *a*

[3] http://www.nestorgames.com/#blue_detail
[4] Personal correspondence from designer Néstor Romeral Andrés.
[5] *Atari* is a term from Go that refers to the immediate threat of capture.

single eye, unlike Go, such as the white group in the lower left corner of Figure 6 (left).

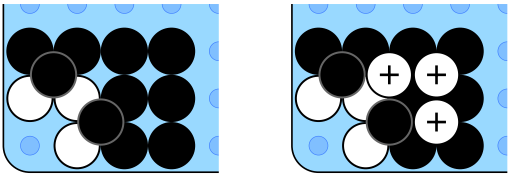

Figure 6. The white group is safe.

Black cannot play in the corner cell as the white zombies would survive the move and mean that the black piece just played would have no freedom, which is not allowed. Instead, White can build from this surprisingly strong base and extend their group to attack further into the board (right). Zombies can be dangerous, and players must weigh the pros and cons of any move that would create enemy zombies very carefully. What at first appeared to be a crippling bug in the game's geometry proved to be a key feature on deeper analysis.

2.4 The L Game

Figure 7 shows the starting position of the L game, in which two players (White and Black) take turns moving their L-piece to occupy four board cells (at least one of them different), then may optionally move one of the two neutral (grey) pieces to an empty cell. A player loses if they cannot move their L-piece on their turn.

The L game was designed almost 50 years ago by psychologist Edward de Bono 'to produce the simplest possible game that could still be played with a high degree of skill' [6, p. 120]. However, it has a serious flaw that would cripple it in the eyes of today's designers: a simple strategy exists that allows players to avoid defeat indefinitely.

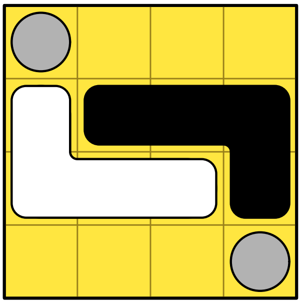

Figure 7. The L game starting position.

This strategy, discovered a decade after the game's invention [2], would seem to eliminate the desired 'high degree of skill', as any player who knew the strategy was as unbeatable as any other. However, the L game still stands as an icon of elegant design [3], has a certain meditative appeal to it, working out the winning strategy can itself be an interesting challenge, and in terms of value for money not many games offer infinite non-trivial play for such simple equipment. So what is clearly a bug in the game could charitably be viewed as a feature in a certain light.

2.5 Reversi

When is a bug not really a bug? Consider the case of Reversi, which most readers should be familiar with, in which two players take turns placing a piece of their colour such that the move caps a line of enemy pieces with a mover's piece at both ends, and the enemy line is then flipped to the mover's colour.

The four corner cells of the board constitute degenerate cases as these are the only cells on which pieces can *never* be flipped once they are placed, as each corner represents a terminal cell of both lines leading into it. For example, the white piece in the top left corner of Figure 8 can never be flipped.

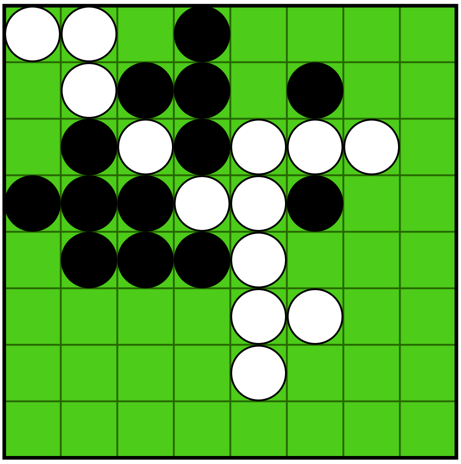

Figure 8. A Reversi position.

While these degenerate cases may be called a bug in some informal sense, this is really stretching the analogy, as they do not violate the game's inherent 'flip if capped' rule; it is simply impossible to cap them. Instead, this makes the corners key strategic points in the game. This is clearly a feature of the game and not a bug.

2.6 Petty Diplomacy

In multi-player games, *petty diplomacy* is the tendency for temporary coalitions to form between players against their common opponent(s). This is described as a serious problem by Schmittberger, to the extent of making many existing three-player games unplayable [7, pp. 44–45]. However, is this really such a problem?

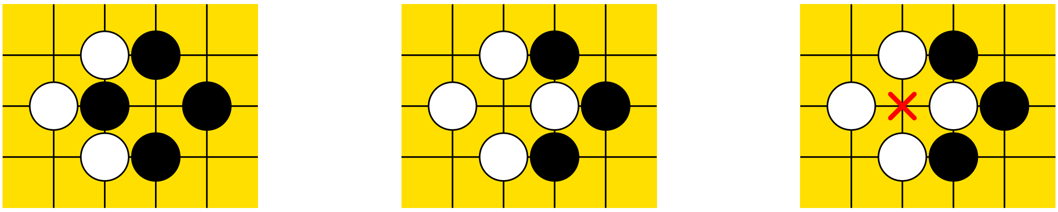

Figure 9. The *ko* rule in Go: Black cannot immediately recapture at the point marked ×.

I make the distinction between *strategic* and *non-strategic* coalitions [14]. Strategic coalitions are those that occur within the framework of the game, for example, when two losing players temporarily cooperate to haul back the leading player, which is also known as the *tall poppy syndrome*. This can actually be cast in a positive light, as it provides a natural balancing mechanism that does not need not be stated in the rules and prolongs the contest. It is not necessarily a bad thing in all cases, provided that a player is eventually able to win by establishing a strong enough position to overcome such alliances.

Non-strategic alliances are those that form outside the game, such as secret pre-arranged agreements between friends against players they do not like. These are obviously anathema to games of strategy and should be eliminated as much as possible.

Many multi-player games include specific rules to minimise the effects of petty diplomacy. However, one notable exception is So Long Sucker! [9], which actually *encourages* strategic coalitions as an integral part of the game and revels in the emergent chaos.

I believe that the dangers of petty diplomacy have been overstated, and that it may actually be a bug for some games but a feature for others. Unfortunately, this relies on players competing in the right spirit and not exploiting non-strategic coalitions.

3 Bug into Feature

This second set of examples includes games with design bugs which have been turned into design features, through judicious rule changes. Note that these are not simple bug fixes, but cases in which some inherent flaw has been turned into a positive feature with a simple twist.

3.1 Go

The surround capture rule in the game of Go raises the danger of infinite cycles in play. For example, the position shown in Figure 9 (left), in which White captures a black stone (middle) but in doing so puts the capturing piece in immediate danger of recapture, which would repeat the board position from the previous turn.

To avoid such cycles, go has a *ko* rule that forbids immediate recapture, so that Black is not allowed to make the move marked × in Figure 9 (right). There are also two different forms of *superko* rule, called *positional superko* and *situational superko*, that forbid the repetition of *any* previous board position.

It would be sacrilege to any serious Go player to suggest that any aspect of the game is not perfect. However, I believe that infinite cycles in play are clearly a bug of the surround capture rule on the square grid,[6] and that the *ko* rule is a bug fix; a very good bug fix, admittedly, as *ko* battles have proven to be a key element of the game, to which significant study has been devoted.

Another solution to this bug is found in Capture Go, also known as Atari Go, in which the first player to make a capture wins the game. This neatly sidesteps the problem by cutting the game short before a cycle can possibly occur, but results in a less deep game. The *ko* rule is probably the most striking case in any board game of a crippling bug turned into a sublime feature, through a simple rule tweak.

3.2 Gonnect

Gonnect is played using the rules of Go, but with two important differences [6]:

1. Players win by connecting their sides of the board with an orthogonally connected chain of their pieces.
2. Players cannot pass.

The first rule makes Gonnect a connection game [10] as much as a territorial game, giving it a unique and interesting character, while the second rule is necessary to fix an inherent bug due to deadlocks.

For example, Figure 10 shows a game in progress that has apparently reached an impasse,

[6] Infinite cycles are much less of a concern on the hexagonal grid [5].

as both players have safe groups and cannot intrude into their opponent's groups due to Go's 'no suicide' rule. However, the fact that players cannot pass means that the mover is forced to fill in one their own eyes, which makes that group vulnerable to capture and puts the mover in a losing position.

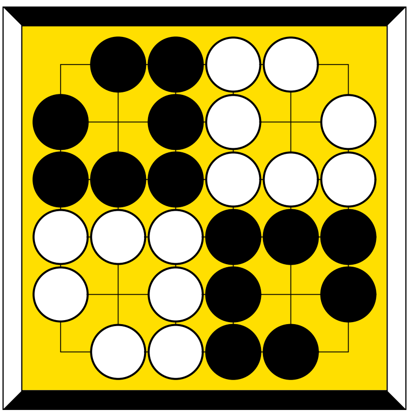

Figure 10. A temporary deadlock in Gonnect.

This 'no pass' rule elegantly solves the problem of deadlocks by simplifying the Go rules, if one considers the absence of passing to be the default case unless passing is specifically allowed.

3.3 Trax

Trax is a tile placement game in which players place tiles in an effort to make a closed path of their colour within a given area limit [10, p. 183]. Its designer, David Smith, encountered the same problem with unplayable spaces as found in Mambo and Blue, but over 40 years earlier.

Consider Figure 11 (left), in which Black has just placed tile *a*. White move *b* (right) would create an unplayable space marked ×, as no tile has three black sides.

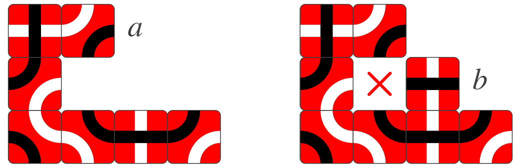

Figure 11. Move *b* would cause an unplayable space.

To fix this problem, Smith added a *forced move* rule stating that if any tile placement creates any positions at which *exactly one* tile can legally be placed, then those tiles *must* be placed there as part of the move, possible triggering further forced moves. This not only addressed the unplayable space problem by making them much rarer in practice, but allowed beautiful sequences of forced moves that add a strategic dimension.

For example, Figure 12 shows how move *a* triggers a sequence of forced moves *b*, *c*, d_1 and d_2 which complete a closed black path to win the game for Black. The forced move rule not only fixed a bug, but created a key feature of the game.

There can be different ways to handle a given bug. While unplayable spaces are embraced in Mambo and Blue as part of the game, here they are greatly reduced by forced moves.

3.4 Chess

Another well-known example is the promotion of pawns in Chess. Consider the position shown in Figure 13, in which White has just moved their pawn to the far row and is about to promote it to a greater piece.[7]

Since pawns can only move forwards, they would otherwise lodge on the far row with no possible moves and play no further part in the game, except perhaps as stationary blockers. Forward pawn movement is the bug and promotion is the fix.

Promotion adds a new dimension to the game, by providing a game-changing discontinuity – the weakest piece suddenly becomes the strongest piece – which makes each pawn a potential time bomb worthy of more respect than mere cannon fodder.

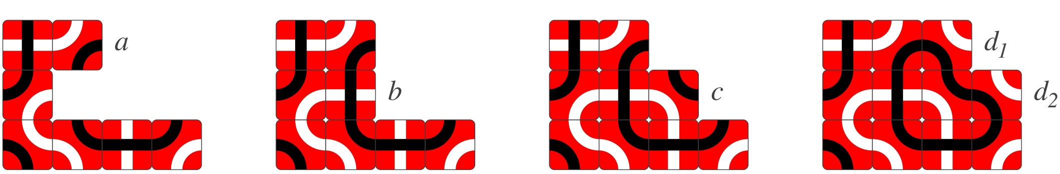

Figure 12. Move *a* triggers forced moves *b*, *c*, d_1 and d_2 which win for Black.

[7]Position from 'How to Play Chess': https://www.youtube.com/watch?v=FJ2CdBsTis4

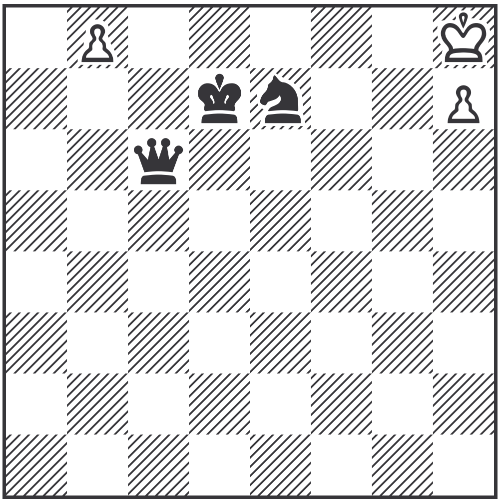

Figure 13. A pawn being promoted... but to what?

Further, the choice of which piece to promote to is not always a given, and can provide an interesting problem in itself. For example, White would lose if they promote their advanced pawn in Figure 13 to the obvious Queen, and instead must *underpromote* it to a lesser piece – a knight in this case – to guarantee victory.

Another solution to the problem of forward-only pawn movement is found in the game of Breakthrough, in which players start with a row of pawns on their home row and win by moving one of their pawns to the far row.[8] Again, the bug is fixed by modifying the win condition to cut the game short as soon as the problem occurs, leading to a much simpler game.

3.5 Andantino

Andantino is another minimalist game designed by David Smith [10, p. 289], in which two players take turns placing a hexagonal tile of their colour such that:

1. Each tile placement is adjacent to at least two existing tiles.

2. A player wins by making a line of 5-in-a-row of their colour, or by completing a connected chain of their colour that encloses at least one enemy tile.

Figure 14 shows the starting position with Black to play (legal moves are indicated by dots). Without the 'two existing tiles' rule, it would be too easy to block line threats in Andantino. For example, move *a* in Figure 15 would easily negate White's most obvious line threat, and the 5-in-a-row goal would play a reduced part in the game.

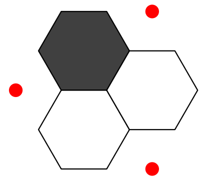

Figure 14. Starting position in Andantino.

The problem is exacerbated by the fact that Andantino is a boardless game, so tiles must grow incrementally upon existing tiles, and cannot be placed some distance away where potential lines might converge, further reducing the likelihood of players actually achieving 5-in-a-row.

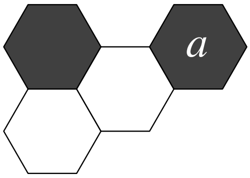

Figure 15. Move *a* would be an easy block.

Most n-in-a-row games are played on boards, where distant threats are typically the key to winning. For example, Figure 16 shows a strong (winning) move for White at the point of intersection of two lines on the square grid in Gomoku. The 'two existing tiles' rule is an elegant fix to this problem, which exploits the underlying geometry of the game to make it harder to block lines.

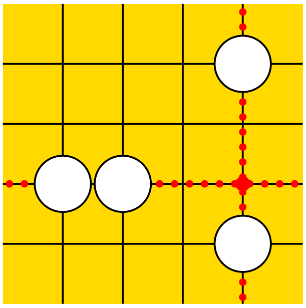

Figure 16. Distant line threats in GoMoku.

There is also a practical need for this rule, as it ensures that the main body of pieces remains clustered together in a tight hexagonal formation, rather than widely spread out where placement errors can accumulate to make the (implied) hexagonal grid hard for players to see. Further, it adds an extra constraint that players can exploit for tactical gain, allowing interesting passages of forced play, and requiring players to plan ahead in order to achieve desired tile placements.

[8] https://en.wikipedia.org/wiki/Breakthrough_(board_game)

For example, Figure 17 (left) shows a position with White to play. White move *a* forces blocking reply *b*, allowing White to play *c*, which sets up a winning fork (indicated +).

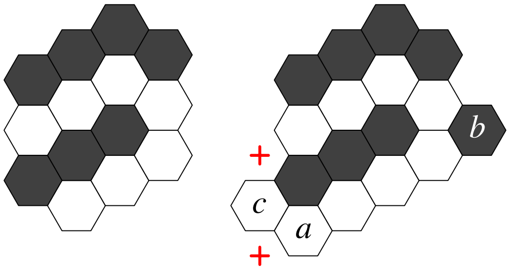

Figure 17. A winning play for White.

The 'two existing tiles' rule fixes the problem of weakened line threats to bring them more into the game, while adding a tactical aspect that requires players to plan ahead more carefully.

3.6 Swap Rule

Figure 18 shows a 9×9 game of Hex, which is played with extremely simple rules: players take turns placing a piece of their colour on an empty cell, and win by connecting the board sides of their colour with a chain of pieces of their colour [11].

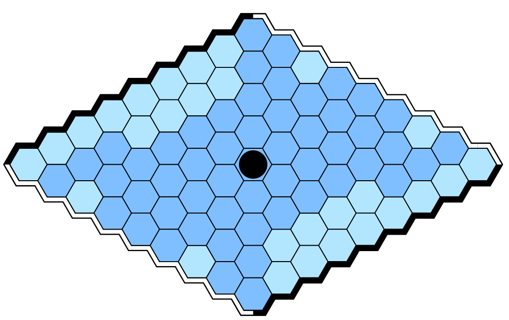

Figure 18. A winning opening in 9×9 Hex.

However, this rule set has a crippling flaw in that the first player has a huge (winning) advantage if allowed an unconstrained first move. For example, Black should win after opening in the centre cell, as shown in the figure, unless they make a serious mistake.

To fix this bug, Hex is played with an additional rule called the *swap rule* or *pie rule*: in reply to the opening move, the second player may elect to swap colours instead of moving. This stops the first player making an overly strong opening move and results in more balanced games. For example, the dark cells shown in Figure 18 are proven to be winning moves for the opening player on the 9×9 board, and should be swapped by White [12].

The swap rule is a somewhat inelegant solution, that ruins the simplicity of this otherwise minimalist rule set, but is a necessary evil if the game is to work between players of similar skill. However, this rather clumsy hack is turned into something of an art form in the game of Unlur.[9]

Unlur is played on a hexagonal grid of hexagons, by two players, White and Black. White aims to connect any two opposite board sides with a chain of white pieces, while Black aims to connect any three non-adjacent board sides with a chain of black pieces. An important twist is that the players are not initially assigned a colour; the game begins with a *contract phase* in which both players place black pieces, until one of them passes to declare themselves Black and their opponent White. Thereafter, players take turns placing a piece of their colour.

For example, Figure 19 shows a typical opening sequence after three moves. This position is probably strong enough that the next player should pass to claim the Black role.

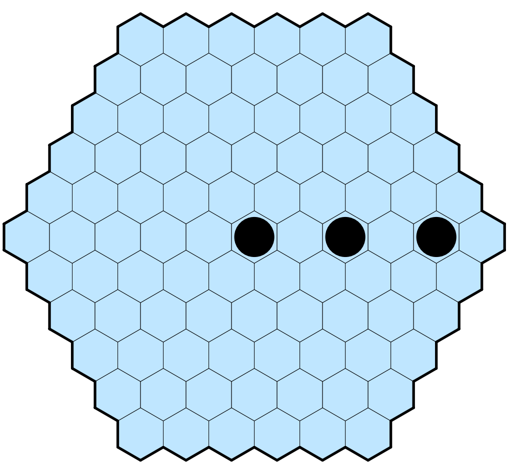

Figure 19. An opening sequence in Unlur.

The swap rule has thus been embraced in Unlur and transformed into a contract phase that neatly balances out this game's very unequal goals. In fact, this contract phase is an innovative feature that defines this game to a large extent; the decision of exactly when to pass is often the most important decision in a game of Unlur. This clumsy bug fix from Hex, rather than being downplayed, has been expanded and seamlessly integrated into the rules in this case.

[9]http://www.di.fc.ul.pt/~jpn/gv/unlur.htm

4 Fuzzing

The sections above describe cases of bugs that turned out to be features, and bugs that were turned into features. However, some bugs remain bugs and have no easy fix; these are the game designs that you do not see. But perhaps even these might have a use.

Returning to the analogy of bugs in computer programming, *fuzzing* is the practice of stress testing software by deliberately introducing bugs into its input, in order to test its resilience to error [13]. Could a similar approach also be taken to game design?

This might entail systematically testing a given rule set with situations known to cause bugs with previous similar rule sets. For example, when designing a tile placement game, the obvious bug to test for is the occurrence of unplayable points, as seen in the Mambo, Blue and Trax examples. The designer should test their proposed rule set with every possible way that such a situation can occur, as a form of boundary value testing. Does the problem occur? If so, does the rule set handle it sufficiently? If not, can it be turned into a feature of the game? Does the problem produce any unexpected emergent behaviour that might inspire further improvements to the game or even entirely new games?

5 Conclusion

The examples presented above show how apparent bugs in a design are not always bad, and, even if they are, can often still be exploited to good effect. The most interesting features of games can emerge from fixing bugs.

It is important to identify bugs in the design process, but equally important to determine whether they actually *are* bugs. Before discarding an iteration of the rule set that appears flawed, ask yourself: is this behaviour a bug or a feature? In either case, how can it be exploited to best effect?

Some of the above examples (Go, Gonnect and Trax) were also included in my article 'Embed the Rules' published in the first issue of *Game & Puzzle Design* [14]. This is not coincidence – or laziness! – as it is an efficient design practice to fix bugs implicitly through the judicious use of rules or geometry, and to incorporate the resulting side effects into the game, where possible.

Acknowledgements

Thanks to Greg Schmidt for discussions, Russ Williams for alerting me to the practice of fuzzing, Stephen Tavener for suggesting pawn promotion in Chess and Néstor Romeral Andrés for the Blue example. This work was supported by a QUT Vice-Chancellor's Research Fellowship.

References

[1] de Bono, E., 'The L Game: Strategic Thinking', *The Five-Day Course in Thinking*, London, Penguin, 1967.

[2] Gijlswijk, V. W., Kindervater, A. P., van Tubergen, G. J. and Wiegerinck, J. J. O. O., *Computer Analysis of E. de Bono's L-Game*, Report #76-18, Department of Mathematics, University of Amsterdam, 1976.

[3] Browne, C., 'Elegance in Game Design', *IEEE Transactions of Computational Intelligence and AI in Games*, vol. 4, no. 3, 2012, pp. 229–240.

[4] Browne, C., *Margo Basics*, Raleigh, Lulu Press, 2013.

[5] Browne, C., 'Go Without Ko on Hexagonal Grids', *ICGA Journal*, vol. 35, no. 1, 2012, pp. 37–40.

[6] Browne, C., 'Gonnect: The Best of Go and Hex', *Abstract Games Magazine*, vol. 1, no. 6, 2001, pp. 12–21.

[7] Schmittberger, R. W., *New Rules for Classic Games*, New York, John Wiley & Sons, 1992.

[8] Browne, C., 'Coalition Control Through Forced Betrayal', *Game & Puzzle Design*, vol. 1, no. 2, 2015, pp. 50–52.

[9] Hausner, M., Nash, J. F., Shapley, L. S. and Shubik, M., 'So Long Sucker, A Four-Person Game', *Game Theory and Related Approaches to Social Behavior*, New York, John Wiley & Sons, 1964, p. 1.

[10] Browne, C., *Connection Games: Variations on a Theme*, Wellesley, AK Peters, 2005.

[11] Browne, C., *Hex Strategy: Making the Right Connections*, Wellesley, AK Peters, 2000.

[12] Pawlewicz, J. and Hayward, R. 'Scalable Parallel DFPN Search', *Computer Games (CG 2013)*, van den Herik, H. J. *et al.* (ed.), LNCS 8427, Springer, 2014, pp. 138–150.

[13] Sutton, M., Greene, A. and Amini, P., *Fuzzing: Brute Force Vulnerability Discovery*, New York, Addison-Wesley, 2007.

[14] Browne, C., 'Embed the Rules', *Game & Puzzle Design*, vol. 1, no. 1, 2015, pp. 60–70.

Cameron Browne is a Vice-Chancellor's Senior Research Fellow at QUT, Brisbane, whose research interests include artificial intelligence and automated game design.
Address: School of EECS, Science and Engineering Faculty, QUT, Brisbane, 4001, Australia.
Email: c.browne@qut.edu.au